In the Claims:

No claims have been added.

Claims 7, 10 and 12—23 are canceled without jeopardizing the Applicant's right to file the same or similar claims in the future.

Accordingly, claims 1—6, 8—9, 11 and 24—25 are pending and are listed as follows:

1. (Currently amended.) A method for processing a database query, comprising:

partially pre-aggregating records in a database according to a single grouping column to provide a result that contains at least two records having like grouping column values; ~~and~~

aggregating records derived from the partial pre-aggregation to provide a result that contains records having unique grouping column ~~values.~~ values; and

estimating the costs and benefits of the partial pre-aggregation, and partially pre-aggregating the records only if the estimating indicates that the benefits are greater than the costs.

2

2. (Original.) The method as recited in claim 1, wherein the partially pre-aggregating further comprises:

maintaining a record store in memory, the record store having one record for each different grouping column value encountered in the operation;

receiving a new record;

combining the new record with a record having the same grouping column value, if such a record exists; and

adding the new record to the record store in the memory if there is no record in the record store that has the same grouping column value as the new record.

3. (Original.) The method as recited in claim 2, further comprising:

adding additional new records to the record store until the record store reaches a capacity such that it can accept no new records; and

outputting one or more records from the record store to a subsequent database operator.

4. (Original.) The method as recited in claim 3, wherein after the one or more records have been output to the subsequent database operator, the adding and outputting are repeated until there are no new records to process.

5. (Original.) The method as recited in claim 4, wherein any records remaining in the record store after there are no new records to process are output to the subsequent database operator.

6. (Original.) The method as recited in claim 3, wherein the subsequent database operator is a join.

7. (Cancelled.)

8. (Original.) The method as recited in claim 1, wherein the partially pre-aggregating includes utilizing a hashing function.

9. (Original.) The method as recited in claim 1, wherein the partial pre-aggregating creates a record store in memory, and wherein the method further comprises utilizing the record store in memory for one or more other database operators.

10. (Cancelled.)

11. (Original) A computer programmed to perform the method recited in claim 1.

12—23. (Cancelled.)

4

24. (Currently amended.) A relational database computer program stored on a computer-readable medium, the relational database computer program comprising computer-executable instructions that, when executed on a computer, perform the following steps:

receiving a stream of input records;

aggregating the input records in the stream according to a single grouping column as it is received to create a record store;

joining records in the record store with other data;

aggregating the records output from the join; ~~and~~

determining if it is optimal to aggregate the input records prior to performing the join;

performing the aggregation prior to the join only if a determination is made that it is optimal to perform an aggregation prior to the join; and

wherein the records output from the join include at least two records that have an identical grouping column value in the single grouping column.

25. (Original.) The relational database computer program as recited in claim 24, wherein:

the record store has a capacity that is less than the number of records in the stream of input records; and

the aggregating each input record is performed until the record store reaches capacity.

26. (Cancelled.)

5